(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 431 434 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **90122513.6**

(51) Int. Cl.5: **G06F 13/42**

(22) Date of filing: **26.11.90**

(30) Priority: **02.12.89 GB 8927317**

(43) Date of publication of application:
12.06.91 Bulletin 91/24

(84) Designated Contracting States:
DE FR GB IT NL

(71) Applicant: **MOTOROLA, INC.**
1303 East Algonquin Road
Schaumburg, IL 60196(US)

(72) Inventor: **Burri, Michel**
32 Ch. Fr. Lehman
CH-1218 GD-Saconnex(CH)

(74) Representative: **Dunlop, Hugh Christopher et al**
Motorola European Intellectual Property Operations Jays Close Viables Industrial Estate
Basingstoke, Hampshire RG22 4PD(GB)

(54) Data interface system.

(57) This invention relates to a data interface system (2) for use with a processor (4) such as a microcomputer unit (MCU) for monitoring and control functions. The data interface system is coupled to a plurality of devices to be monitored and controlled and comprises:

a plurality of interface means (6a-h) for coupling to said plurality of devices wherein each one of said interface means comprises first (24) and second (20) data latch means, said first data latch means for holding data pushed to said plurality of interface means by said processor to be selectively outputted to said plurality of devices, said second data latch means for holding data received from said plurality of devices to be pulled to said processor; and

data communication means coupled to said processor and to said plurality of interface means being arranged to initiate in response to said processor a push/pull operation comprising a first sequence wherein a data signal is pushed from said processor to said plurality of interface means to be held in the first data latch means (24) of a selected one of said plurality of interface means for outputting to one of said plurality of devices and a subsequent second sequence wherein a data signal is pulled from the second data latch means (20) of said selected one of said plurality of interface means to said processor.

The data communication means may comprise any one of the following configurations: one bus line (8) for transmitting data and initiating a push/pull operation; one push/pull initiation line (210) and one bus line (212); one push/pull initiation line (321) and a bus line (312a-h) for each of the plurality of interface means;and, one push/pull initiation line (410), one bus line (414) coupling the interface means in a chain and a clock line (412) for synchronising the push/pull operation.

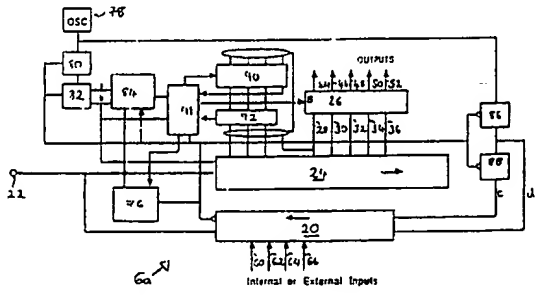


FIGURE 2

EP 0 431 434 A2

BNSDOCID: <EP 0431434A2>

DATA INTERFACE SYSTEM

This invention relates to a data interface system for use with a microcomputer unit (MCU) in control applications, such as automobile engines.

A control system in an automobile engine generally comprises a controlling microcomputer unit (MCU) which can communicate with a plurality of interface devices. The interface devices are coupled to operating parts of the engine such as stepping motors which can control, for example the air-conditioning and ambient temperature in the car, so that these operating parts may be monitored and controlled by the MCU. The interface devices store data received from the MCU which can be outputted to the associated operating parts of the engine. The interface devices also store data received from the associated operating parts of the engine which can be selectively outputted to the MCU.

Known systems which have peripheral devices coupled to a MCU, such as SPI and SCI systems, use serial buses to interconnect the MCU and interface devices wherein the MCU is coupled to each of the interface devices via clock, data and enable lines or transmit and receive lines respectively. When the MCU writes data to an interface device in order to control a particular part of the engine the address of the particular interface device is defined and sent by the MCU followed by a write bit and data bits.

When the MCU reads data from an interface device, the address of the particular interface device is defined and sent by the MCU followed by a read bit whereby the MCU reads data from the addressed interface device.

In order that a read or write operation can be performed, a stop bit or an idle time must be sent on the bus before a read or write operation is initiated.

Such a control system has a major disadvantage in that complex protocol specifying the type of operation is required to perform a write operation followed by a read operation. In addition, a considerable execution time is required for each operation which may be detrimental to the operation of the control system. For example, by the time a write operation has been performed and a read operation initiated if the data read from one of the interface devices required immediate controlling action by the MCU, the MCU could 'miss' reading the data in time and so will be unable to efficiently control an operating part of the engine.

In accordance with the present invention there is provided a data interface system coupled to a processor for monitoring and control functions and to a plurality of devices to be monitored and controlled, said data interface system comprising:

a plurality of interface means for coupling to said plurality of devices wherein each one of said interface means comprises first and second data latch means, said first data latch means for holding data pushed to said plurality of interface means by said processor to be selectively outputted to said plurality of devices, said second data latch means for holding data received from said plurality of devices to be pulled to said processor; and

data communication means coupled to said processor and to said plurality of interface means being arranged to initiate in response to said processor a push/pull operation comprising a first sequence wherein a data signal is pushed from said processor to said plurality of interface means to be held in the first data latch means of a selected one of said plurality of interface means and a subsequent second sequence wherein a data signal is pulled from the second data latch means of said selected one of said plurality of interface means to said processor.

In a preferred first embodiment of the invention the data communication means preferably comprises one bus line coupled to each of the plurality of interface means for outputting to one of said plurality of devices. The processor initiates a push/pull operation by generating and sending for example ten data bits on the bus line. These are detected by a push/pull function detector of each of the interface devices so that the push/pull operation is initiated in each of the plurality of interface devices.

Alternatively, in a second embodiment the communication means may comprise a push/pull line and a data line whereby the push/pull operation is initiated when the processor generates and sends a signal to each of the interface devices on the push/pull line. In this embodiment data is then transmitted on the data line but the push/pull operation is controlled by the signal on the push/pull line.

For these two embodiments an interface device is selected using a bit address generated by the processor and comparing it with a pre-defined address assigned to each of the interface devices.

In a third embodiment in accordance with the present invention the communication means may comprise a push/pull line and a data line for each of the interface devices. Thus, this eliminates the need for an assigned pre-defined address since the processor may select one of the interface devices by selecting one of the data lines.

In a fourth embodiment in accordance with the invention the communication means comprises a push/pull line, a data line intercoupled between the

BNSDOCID: <EP 0431434A2>

interface devices and the processor arranged in a chain and a clock line. The data signal is transmitted through the chain of interface devices from the processor during a push sequence and a data signal is subsequently pulled back through the chain of interface devices from the interface devices during a pull sequence to the processor. An interface device is selected according to a signal on the clock line.

Thus, it will be appreciated that each of the data interface systems according to the present invention described herein, enable a processor in one complete cycle to select an interface device, write data to the selected interface device and then read data stored in the interface device. Thus, the execution time required to write data to an interface device and to then read data from an interface is considerably reduced. This reduces the chances of the controlling MCU 'missing' an event which requires immediate controlling action.

Four data interface systems in accordance with the invention will now be described by way of example only with reference to the accompanying drawings in which:

Figure 1 shows a simplified block schematic diagram of a first data interface system in accordance with the present invention;

Figure 2 shows a block schematic diagram of one of the interface devices of the data interface system of Figure 1;

Figure 3a is a diagrammatic representation of a strobe signal internally generated by the interface device of Figure 2;

Figure 3b is a diagrammatic representation of a clocked signal generated by the interface device of Figure 2;

Figure 3c is a diagrammatic representation of a data signal transmitted on a bus of the data interface system of Figure 1;

Figure 4 shows a simplified block schematic diagram of a second data interface system in accordance with the present invention;

Figure 5 shows a block schematic diagram of one of the interface devices of the data interface system of Figure 4;

Figure 6 shows a simplified block schematic diagram of a third data interface system in accordance with the present invention; and

Figure 7 shows a block schematic diagram of one of the interface devices of the data interface system of Figure 6;

Figure 8 shows a simplified block schematic diagram of a fourth data interface system in accordance with the present invention; and

Figure 9 shows a block schematic diagram of one of the interface devices of the data interface system of Figure 8.

Figure 1 shows a data interface system 2 which is used, in a preferred embodiment, for controlling and monitoring a plurality of stepping motors (not shown) in a car engine. The data interface system 2 comprises a microcomputer unit (MCU) 4, a plurality of interface devices 6a-h and an MCU bus 8. The MCU bus 8 couples each of the interface devices 6a-h to the MCU 4. Although the preferred embodiments will be described in relation to eight interface devices coupled to eight stepping motors it will be appreciated by a person skilled in the art that the invention is not limited thereto.

Each one of the interface devices 6a-h has two bus lines 10a-h and 11a-h for coupling to the stepping motors (not shown). Data is transmitted from the interface devices 6a-h to the stepping motors along buslines 10a-h and data is received from the stepping motors along bus lines 11a-h.

The data interface system 2 in accordance with the invention is arranged to communicate with the MCU 4 by way of a push/pull mode of operation whereby in one complete cycle data can be written to a selected one of the interface devices 6a-h and then immediately after data can be read from the selected one of the interface devices: that is the push sequence is immediately followed by a pull sequence.

The push/pull mode of operation of the data interface system 2 will be described in more detail below, with reference to Figure 2 which shows one of the interface devices, for example interface device 6a.

The interface device 6a comprises an input/output node 22 for coupling to the MCU bus 8, a serial in/parallel out register (SPR) 24 coupled to the input/output node 22 and a parallel in/serial out register (PSR) 20 also coupled to the input/output node 22.

The SPR 24 receives data serially from the MCU bus 8 via the input/output node 22 and subsequently outputs the data in parallel to latches 26 for storage via five parallel bus lines 28-36. Latches 26 have outputs 44-52 which correspond to the bus lines 28-36 and which form the data bus 10a of Figure 1 for coupling to associated stepping motor (not shown).

The PSR 20 has four parallel inputs 60-66 which form the data bus 11a of figure 1 for coupling to one of the stepping motors (not shown). The PSR 20 stores data received in parallel from the associated stepping motor (not shown) via the parallel input lines 60-66. The data is then outputted serially onto the MCU bus 8 via input/output node 22 during a pull sequence.

In the preferred embodiment described herein data on the MCU bus 8 is coded according to the BI-phase code for security and to increase the signal-to-noise ratio. A BI-phase and noise detector 84 detects the BI-phase coding of the transmitted

BNSDOCID: <EP 0431434A2>

data and also the noise on the MCU bus 8. If the coding is correct and the noise is within the systems' limits, a data transmission validation signal is outputted to a validation control circuit 91. If a bit error is detected in the transmitted data, an error signal is generated and outputted to the validation control circuit 91 for storage until the end of the push sequence. By using BI-phase coding, the MCU 4, in order to initiate a push/pull mode of operation, need only generate a start bit to violate all bits present on the MCU bus 8 and hence force the MCU bus 8 to a different logic state. The change in logic state of the MCU bus 8 is detected by push/pull function detectors 76 of each of the interface devices which then executes the push/pull mode of operation of the interface devices. It will be appreciated, however, that different codes may also be implemented.

Each of the interface devices 6a-h has a specific pre-determined address which is stored in a programmed address memory 90. Since there are eight interface devices each address will comprise three bits. To select a particular one of the interface devices 6a-h, the MCU 4 defines the specific address of the interface device to be selected and sends it on the MCU data bus 8. A comparison of the sent addressed and the pre-determined address is then performed in each of the interface devices using a comparator 92.

A programming procedure is required to individually program each interface device with a specific address. This can be executed by the manufacturer or by the end user of the system. It will be appreciated that other methods of selecting one of the interface devices could also be used. For example, a method using a selection of pins using jumpers or thumble switches could also be implemented.

The interface device 6a also comprises an internal oscillator 78 which provides a clock signal as shown for example in Figure 3b for clocking the push/pull mode of operation of the interface device 6a. Different ceramic resonator frequencies may be used and are automatically selected by the MCU 4 in response to the possible bit rate frequency used by the data interface system.

During a push sequence of the push/pull mode of operation, the internal oscillator 78 clocks two counters 80 and 82 generating two separate clock signals a and b. These two clock signals a and b are used by the BI-phase code and noise detector 84 to sample each half bit (or half time slot) of the BI-phase coded data sent by the MCU 4. Depending on the data transmission either an error signal or a data transmission validation signal is outputted to the validation control circuit 91 for storage until the end of the push sequence.

During the pull sequence of the push/pull mode of operation, the internal oscillator 78 clocks two other counters 86 and 88. The clock signal d generated by the counter 86 is used to clock the outputting of the data stored in the PSR 20 to the MCU bus 8. The clock signal c is used to output a signal having a frequency which is a sub-multiple of the frequency of the signal generated by the internal oscillator 78.

Referring now also to Figures 3a and 3c, the manner in which data is communicated between the MCU 4 and the interface devices 6a-h during a push/pull mode of operation will now be described.

In order to write data to the interface devices 6a-h, the MCU initiates a push sequence of the push/pull mode of operation by generating a start bit, or as in the preferred embodiment three start bits 100, which violate the MCU bus 8 forcing the bus to a different logic state.

The push/pull function detectors 76 of each of the interface devices 6a-h detect the change in logic state of the bus whereby the push/pull mode of operation of each of the interface devices is enabled. Control logic (not shown) of the MCU 4 is then set to push data, coded according to the BI-phase code, to the interface devices 6a-h.

Figure 3c shows an exemplary data signal transmitted by the MCU bus 8 during a push/pull mode of operation. The three start bits 100 are followed by nine BI-phase coded bits comprising a sync bit 101 for synchronising the data transmission to the interface devices according to the internal oscillator 78, three address bits 102, 103, 104 to select one of the eight different interface devices 6a-h and five data bits 105-109 to control the five outputs 44-52 of the latches 26 of the selected interface device.

Each of the interface devices 6a-h, once enabled, receives the address and data bits transmitted by the MCU bus 8. The BI-phase and noise detector of each of the interface devices samples the data and if the data transmission is valid outputs a data transmission validation signal to the verification control circuit. The received three bit address is compared with the pre-programmed addresses stored in the programmed address memories 90. When a match of addresses is detected by one of the comparators, a match signal is outputted from the comparator 92 of the selected interface device to the validation control circuit 91.

The push sequence is clocked according to the internal oscillator 78 which is arranged so that number of internal clock signals a and b correspond to the number of address and data bits so as to ensure that there are sufficient clock cycles to push all the data into the SPR. Thus, the number of address and data bits are counted according to the internal clock signals a and b so as to prepare the push/pull function detector 76 to commute from

BNSDOCID: <EP 0431434A2>

the push mode to the pull mode. This commutation is executed in response to the next bit, a pull sync bit 121, following the address and data bits on the MCU bus 8.

A strobe signal S (shown in Figure 3a) is generated by the verification control circuit 91 in response to the pull sync bit 121 if and only if a match of addresses has been detected and if the BI-phase and noise detector validates the data transmission. The strobe signal S is sent to the latches 26 so as to store the contents of the SPR 24 to the latches 26 via lines 28-36. If an error signal is stored in the validation control circuit, the generation of the strobe signal is inhibited so that no data can be stored in the latches 26. The other seven unselected interface devices hold data bits previously stored in their respective latches until such time as the interface device is selected and a strobe pulse is generated for that interface device.

A push sequence typically takes 475 microseconds with the start bits requiring an additional 75 microseconds. Once a push sequence has been completed the push/pull function detector 76 of the interface devices is reset to initiate a pull sequence during which the data stored in the PSR of the addressed interface device is pulled to the MCU 4 via the MCU bus 8 according to the clock signal d.

The data on the MCU bus 8 comprises a pull sync bit 121 for synchronising the data transmission to the MCU 4 according to the internal oscillator 78. The data from the PSR of the addressed interface device comprises three data bits 122 which indicate the internal status of the addressed device. This may concern for example the status of the MCU bus' transmission, the behaviour of the output driver stage of the stepping motor, the nature of the stepping motor connections, or validation of a newly programmed address in the PROM address memory of the interface device. The status data bits 122 are followed by a submultiple frequency signal 125 which in the preferred embodiment described herein is the internal oscillator frequency divided by sixteen. Once the required clock cycles for pulling the data from the PSR of the addressed interface device are completed, the MCU 4 can initiate another push/pull mode of operation in which another of the interface devices is addressed.

The sub-multiple frequency signal 125 allows the MCU 4 to verify whether the data it receives from the interface device is valid and hence whether the MCU bus 8 is working correctly. In addition, the submultiple frequency signal 125 is an indication of the internal oscillator's frequency which can change with time. Thus, the MCU can immediately detect any frequency changes in the clock signal and adjust its frequency accordingly so that the push/pull mode of operation will not be forced out of synchronisation.

When the MCU 4 is pushing data to the SPR of the interface devices, the PSR of all the interface devices continuously capture their own input information via the parallel input lines 60-66 which may be from external input/output pins (not shown) or may be internal status bits.

Referring now to Figure 4, a second data interface system 200 incorporating the invention is similar to the first data interface system described with reference to Figures 1, 2 and 3a-c in that the data interface system 200 is arranged to communicate with an MCU 204 by way of a push/pull mode of operation. However, the data interface system 200 uses two lines, a push/pull line 210 and a data line 212, for coupling the MCU 204 to the plurality of interface devices 206a-h. The data interface system 200 uses a signal generated by the MCU 204 and sent on the push/pull line 210 to execute the push/pull mode of operation in each of the interface devices 206a-h.

Referring also to Figure 5, the structure and function of the interface devices 206a-h, only one of which 206a is shown in detail in Figure 5, are similar to that of the interface device 6a and like components to those of Figure 2 are referenced by the same reference numeral plus the number two hundred. Since a signal on the push/pull line 210 initiates the push/pull mode of operation in each of the interface devices 206a-h, the need for a push/pull function detector 76 is eliminated.

The push/pull line 210 is coupled via a node 225 to the PSR 220, the validation control circuit 291, the BI-phase and noise detector 284 and the internal oscillator counters 280, 282, 286 and 288.

Once a push/pull mode of operation has been initiated by a signal on the push/pull line 210, a data signal is pushed by the MCU 204 to the plurality of interface devices 206a-h on the data line 212. The data signal comprises three address bits to select one of the interface devices 206a-h, followed by data bits.

The operation of the data interface system 200 is identical to that of the data interface system 2 described in detail above. The operation is clocked by the internal oscillator 278 and by resetting the signal on the push/pull line 210 a pull sequence can be initiated. Thus, in one complete cycle data can be pushed to a selected one of the interface devices and subsequently data can be pulled from the selected one of the interface devices.

It will be appreciated that since no start bit(s) is required to enable the push/pull mode of operation the length of the signal transmitted on the data line 212 is shorter than the corresponding signal on the MCU bus 8.

Referring now to Figure 6, a third data interface system 300 incorporating the invention is similar to

BNSDOCID: <EP 0431434A2>

the first data interface system 2 and the second data interface system 200 in that the data interface system 300 is arranged to communicate with an MCU 304 by way of a push/pull mode of operation using two lines 321 and 312. However, each of the plurality of interface devices 306a-h use a dedicated data line 312a-h to communicate with the MCU 304 and a push/pull line 310 to execute the push/pull mode of operation in each of the interface devices 306a-h. Referring also to Figure 7, the structure and function of the interface devices 306a-h, only one of which 306a is shown in Figure 7, are similar to that of the interface devices 206a-h except that the interface devices 306a-h do not require a programmed address memory 90 nor a comparator 92 in order to select one of the interface devices. Like components to those of Figure 2 are referenced by the same reference numeral plus the number three hundred.

To initiate a push/pull mode of operation, a signal generated by the MCU 304 is sent on the push/pull line 310 to each of the interface devices. The MCU 304 selects one of the data lines 312a-h and pushes a data signal to the selected one of the interface devices via the selected line. The MCU 304 resets the signal on the push/pull line 310 so as to initiate a pull sequence whereby data is pulled from the PSR 320 of the selected interface device to the MCU via the selected data line.

Thus, the fifth data interface system 300 does not require a defined bit address in order to select one of the interface devices 306a-h and hence the length of the signal transmitted on one of the data lines 312a-h is shorter than the corresponding signal on the data line 212. Typically the push sequence takes 275 microseconds. The unselected interface devices receive undefined coded data which is not BI-phase coded. For example, data (half time slot) comprising 11111 or 00000. On receipt of such coded data the BI-phase code detector 384 will inhibit the strobe pulse and hence no new data will be stored in latches 326.

Referring now to Figure 8, a fourth data interface system 400 incorporating the invention is similar to the first data interface system 2 in that the data interface system 400 is arranged to communicate with an MCU 404 by way of a push/pull mode of operation. However, the data interface system 400 comprises three lines, a push/pull line 410, a clock line 412 and a data line 414, for coupling the MCU 404 to the plurality of interface devices 406a-h. The interface devices 406a-h are coupled together in a chain via the data line 414 which also couples the first interface device 406a to the MCU.

The data interface system 400 uses a signal generated by the MCU 404 and sent on the push/pull line 410 to execute the push/pull mode of operation in each of the plurality of interface devices 406a-h as with the second and third data interface systems 200 and 300 respectively. However, unlike the three systems 2, 200 and 300 described above, the push/pull mode of operation of each of the interface devices is synchronized by a signal on the clock line 412 coupled to each of the interface devices 406a-h. The interface devices 406a-h therefore do not require their own internal oscillator.

During a push sequence the MCU pushes data through the chain of interface devices. An interface device is selected according to the clock signal on the clock line 412 which is dependent on the number of interface devices in the chain and the cycle time required to push data through an interface device in the chain.

One or more of the interface devices may be updated in the same push mode of operation. Thus, during a push sequence selected interface devices may have their data updated whereas the unselected devices will receive the same data previously stored in the latches 426.

Figure 9 shows a block schematic diagram of one of the interface devices 406a of the data interface system 400. Like components to those of Figure 2 are referenced by the same reference numeral plus the number four hundred.

The interface device 406a comprises a SPR 424 having an input coupled to a data input/output node 421 via a first switch arrangement 427 for receiving a data signal serially from said processor and a PSR 420 having an output also coupled to the data input/output node 421 via the first switch arrangement 427 for outputting serially a data signal. Depending on the state of the first switch arrangement the data input/output node can be switched either to the input of the SPR 424 or the output of the PSR 420. The data input/output node 421 is coupled to the data line 414 for coupling to the next interface device in the chain or if it is the first interface device in the chain to the processor 404.

As with the SPR 24 of the interface device 2 of the first embodiment, the SPR 424 is coupled to latches 426 having parallel bus lines 444-452 which form the data bus 410a of Figure 8. The SPR 424 has an output coupled to an data output/input node 431 via a second switch arrangement 429.

The PSR 420 has four parallel inputs 460-466 which form the data bus 411a of figure 8. The PSR 420 has an input coupled to the data output/input node 431 via the second switch arrangement 429. Depending on the state of the second switch arrangement 429 the data output/input node can be switched to either the SPR's output or the PSR's input.

The push/pull line 410 is coupled via a node

BNSDOCID: <EP 0431434A2>

425 to the first and second switch arrangements 427 and 429 and to the SPR and PSR.

If the MCU 404 initiates a push sequence, the signal on the push/pull line switches the first switch arrangement so that the data input/output node 421 is coupled to the input of the SPR 424 and switches the second switch arrangement so that the data output/input node 431 is coupled to the output of the SPR for each of the interface devices 406a-h. Thus, data is pushed from the MCU 404 through the SPRs of the chain of interface devices according to the clock signal.

The clock signal is provided so that the interface device to be selected receives a special signal on the clock input 423 by the time the data is pushed to its' data input/output node 421. On receipt of the special clock signal the data instead of being outputted from the data output/input node 431 to the next interface device is latched in the latches 426. On the next clock cycle the signal on the push/pull line 410 is reset so that a pull sequence is initiated.

The MCU sends a predetermined number of clocks which correspond to the number of bits implemented in the chain interface, after which the push/pull line 410 is reset so that a pull sequence is initiated.

During a pull sequence the signal on the push/pull line 410 switches the first switch arrangement 427 so that the data input/output node 421 is coupled to the output of the PSR 420 and switches the second switch arrangement 429 so that the data output/input node 431 is coupled to the input of the PSR 420. Thus, data held in the PSR of the chain of interface devices can be pulled through the PSRs to the MCU 404.

Many different data interface systems incorporating the push/pull mode of operation may be created. An advantage of using two or three lines to interconnect the MCU with the interface devices, as in the second, third and fourth embodiments of the invention, as opposed to the one line of the first embodiment is that the operating protocol is simpler. Also, the additional lines of the third and fourth embodiments eliminate the need to use a programming procedure to program the address into the programmed address memory. However, a major advantage of the first embodiment is that it uses one bus line to couple the MCU to a plurality of interface devices.

It will be appreciated that each of the data interface systems according to the present invention described herein, enable the MCU in one complete cycle to select an interface device, write data to the selected interface device and then read data stored in the interface device. Thus, the execution time required to write data to an interface device and to then read data from an interface is considerably reduced. This reduces the chances of the controlling MCU 'missing' an event which requires immediate controlling action.

It will also be appreciated that the bus line or lines of the preferred embodiments which are required to couple the interface devices to the MCU are considerably cheaper than the buses used in the prior art systems.

It will be appreciated that another advantage of the data interface systems according to the present invention over the prior art systems is that the data interface systems according to present invention requires a simple operating protocol which can be controlled by software. Thus, the present invention can be used with any MCU available on the market and will only need to be programmed accordingly on initialisation.

Claims

1. A data interface system coupled to a processor for monitoring and control functions and to a plurality of devices to be monitored and controlled, said data interface system comprising:

   a plurality of interface means for coupling to said plurality of devices wherein each one of said interface means comprises first and second data latch means, said first data latch means for holding data pushed to said plurality of interface means by said processor to be selectively outputted to said plurality of devices, said second data latch means for holding data received from said plurality of devices to be pulled to said processor; and

   data communication means coupled to said processor and to said plurality of interface means being arranged to initiate in response to said processor a push/pull operation comprising a first sequence wherein a data signal is pushed from said processor to said plurality of interface means to be held in the first data latch means of a selected one of said plurality of interface means for outputting to one of said plurality of devices and a subsequent second sequence wherein a data signal is pulled from the second data latch means of said selected one of said plurality of interface means to said processor.

2. A data interface system according to claim 1 wherein said communication means comprises one bus line coupled to each one of said plurality of interface means.

3. A data interface system according to claim 2 wherein said data signal pushed by said processor on said one bus line comprises at least

BNSDOCID: <EP 0431434A2>

one start bit for initiating a push/pull operation, address bits for selecting one of said plurality of interface means and at least one data word.

4. A data interface system according to claim 3 wherein each one of said plurality of interface means further comprises a push/pull operation detector for detecting said at least one start bit for initiating a push/pull operation and in response thereto initiating the push/pull operation in each one of said plurality of interface means.

5. A data interface system according to claim 3 or 4 wherein each one of said plurality of interface means further comprises:

programmable address means for storing pre-defined address bits specific to each one of said plurality of interface means;

a comparator for comparing the stored address with said address bits of the pushed data signal and for generating in response to said comparator detecting a match of addresses an activating signal whereby the selected one of said plurality of interface means is activated to receive and hold in its first data latch means the at least one data word following said address word of said data signal during a first sequence.

6. A data interface system according to any preceding claim wherein each one of said plurality of interface means further comprises a clock signal generating means for providing a clock signal for synchronising the push/pull operation.

7. A data interface system according to claim 6 wherein said clock signal generating means is arranged to provide a sub-multiple frequency signal having a frequency which is a sub-multiple of the frequency of said clock signal.

8. A data interface system according to claim 7 wherein said data signal pulled from said selected interface device means comprises the data received by said second data latch means from said plurality of devices and said sub-multiple frequency signal, said sub-multiple frequency signal allowing for said processor to verify that the pulled data information is valid.

9. A method of controlling and monitoring a plurality of devices using a processor coupled to said plurality of devices by a data interface system, said data interface system comprising a plurality of interface means and communication means coupled to said processor and to said plurality of interface means, each one of said plurality of interface means comprising first and second data latch means, said first data latch means for holding data pushed to said plurality of interface means by said processor for writing to said plurality of devices, said second data latch means for holding data received from said plurality of devices to be pulled to said processor, said method comprising a first sequence followed by a subsequent second sequence, said first sequence comprising the steps of:

initiating a push/pull operation in each of said plurality of interface means;

selecting one of said plurality of interface means and activating said selected one of said plurality of interface means to receive a data signal pushed by said processor via said communication means; and

holding said data signal in said first data latch means of said selected one of said plurality of interface means for writing to one of said plurality of devices, said second sequence comprising the steps of:

pulling a data signal from said second data latch means to said processor via said communication means, said data signal comprising data received by said second data latch means from said one of said plurality of devices.

BNSDOCID: <EP 0431434A2>

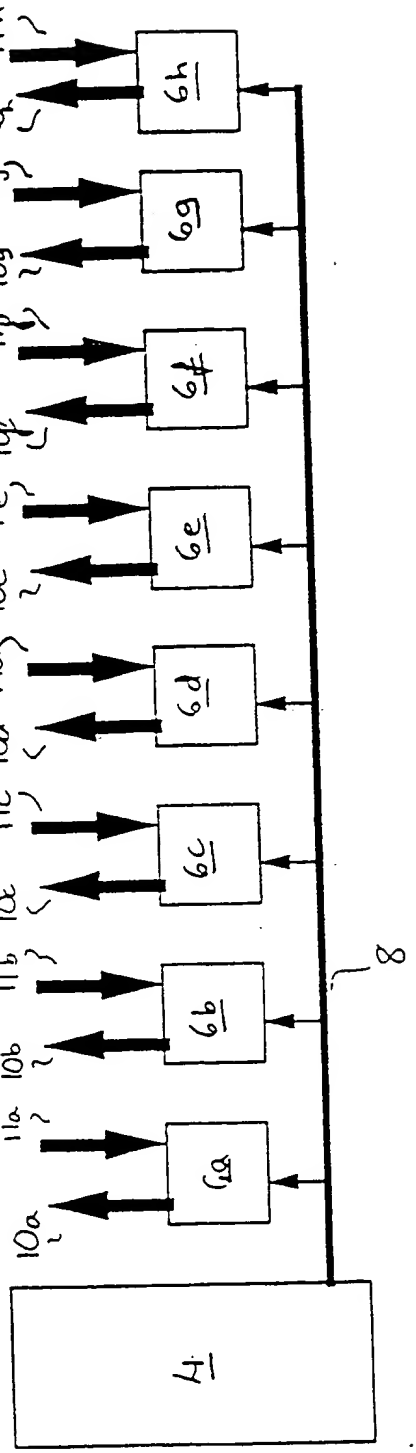


Figure 1

2

BNSDOCID: <EP 0431434A2>

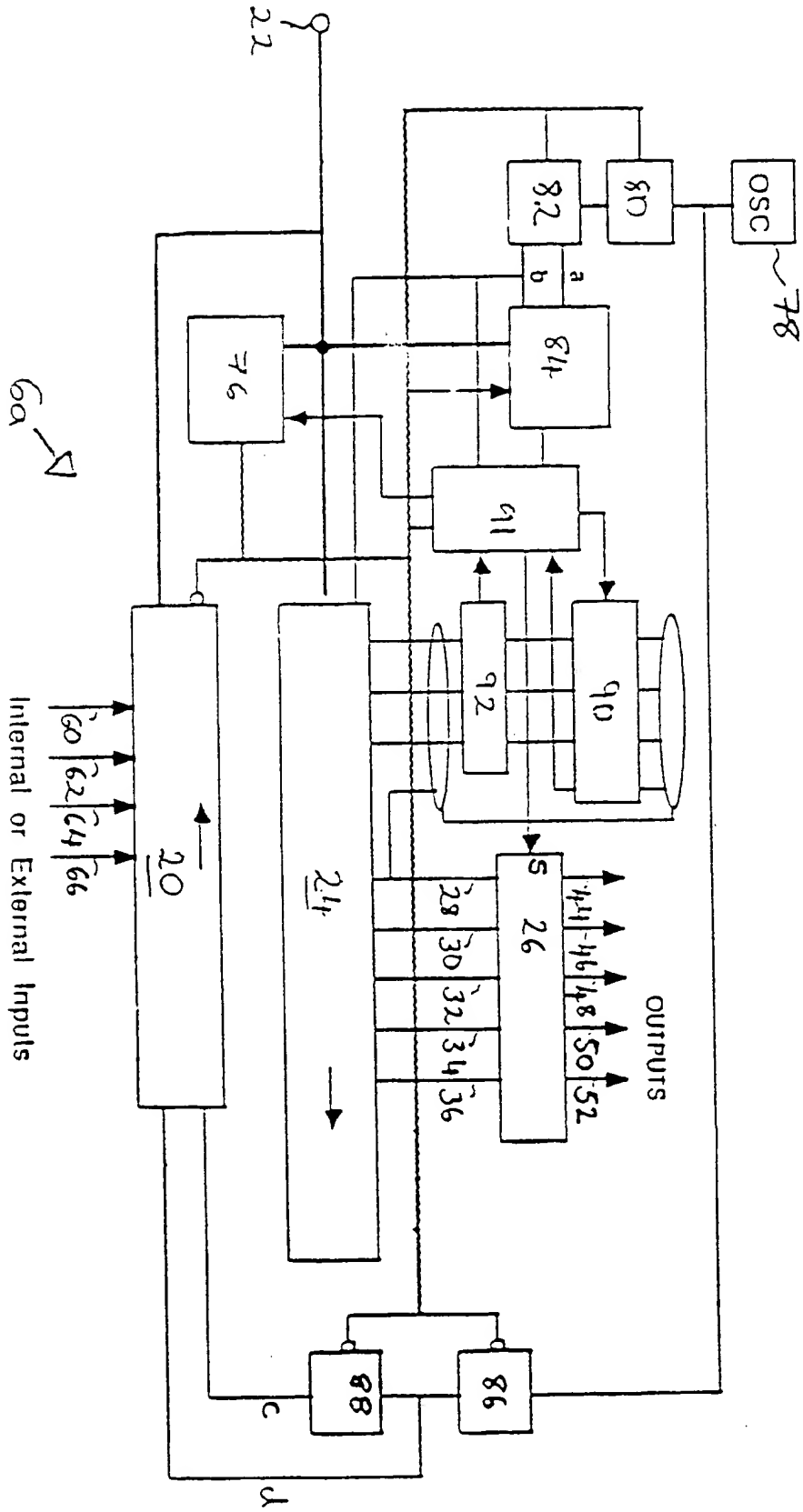


FIGURE 2

BNSDOCID: <EP 0431434A2>

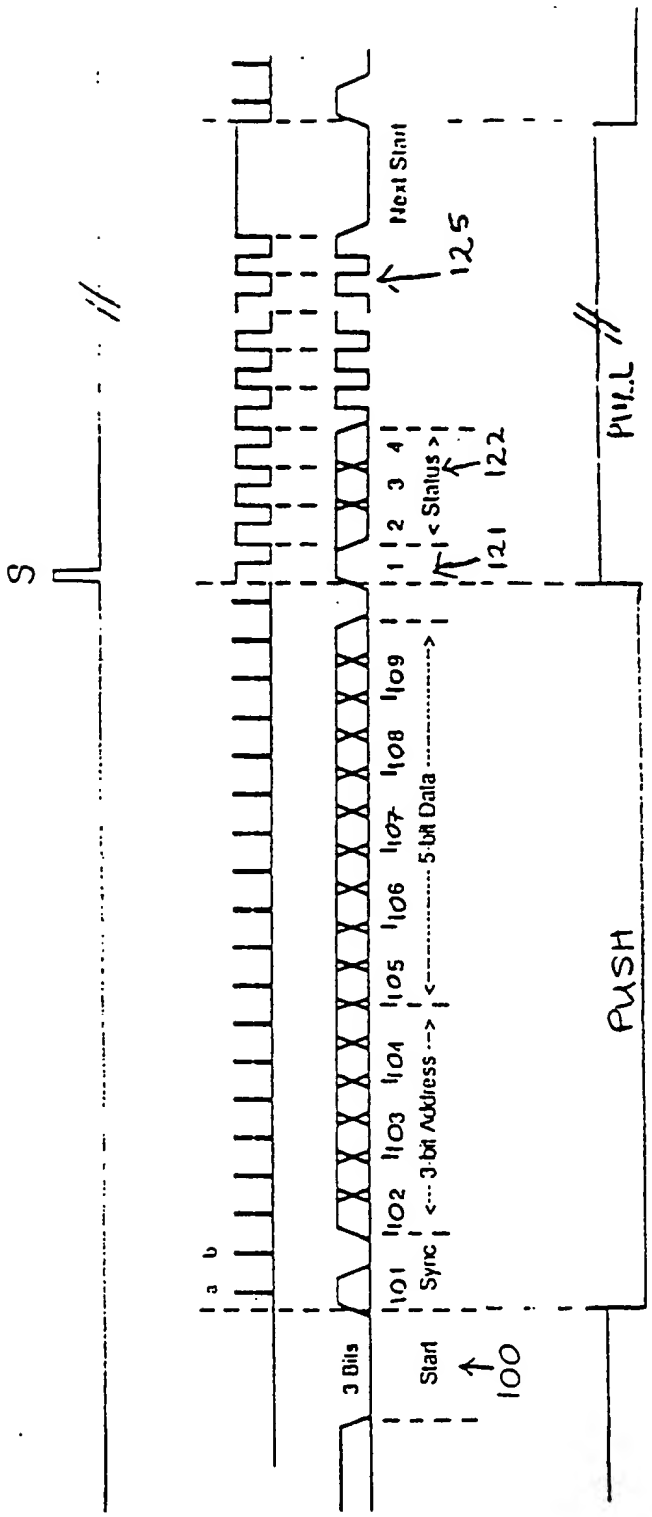


FIGURE 3a

FIGURE 3b

FIGURE 3c

BNSDOCID: <EP 0431434A2>

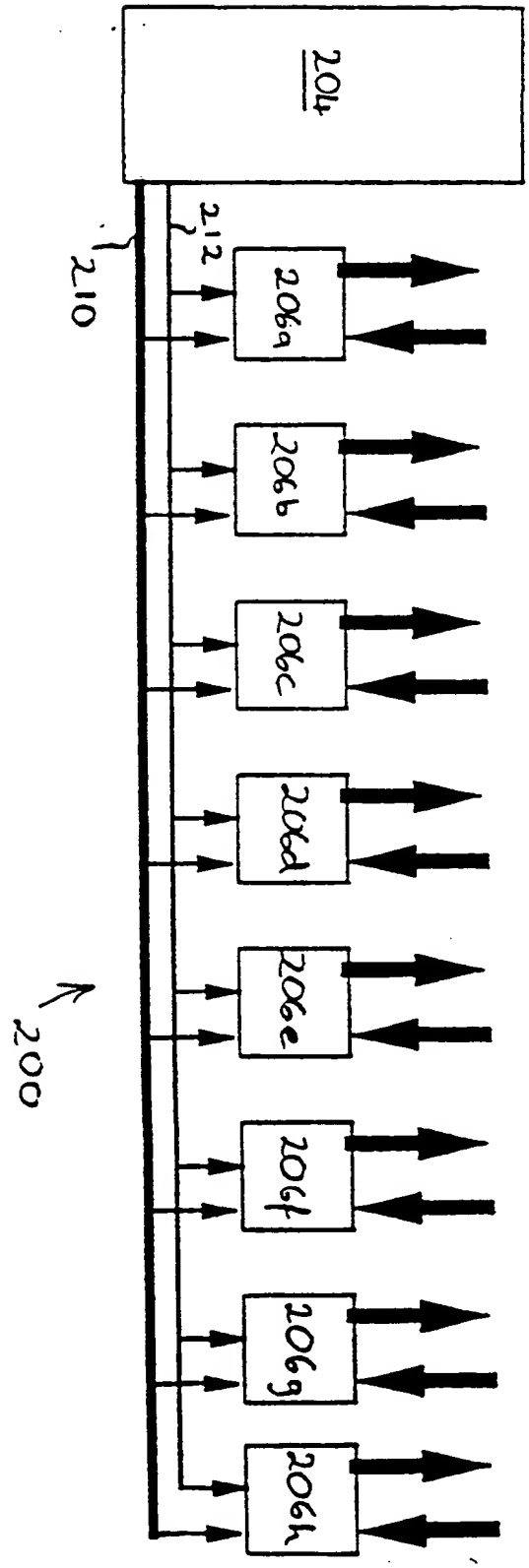
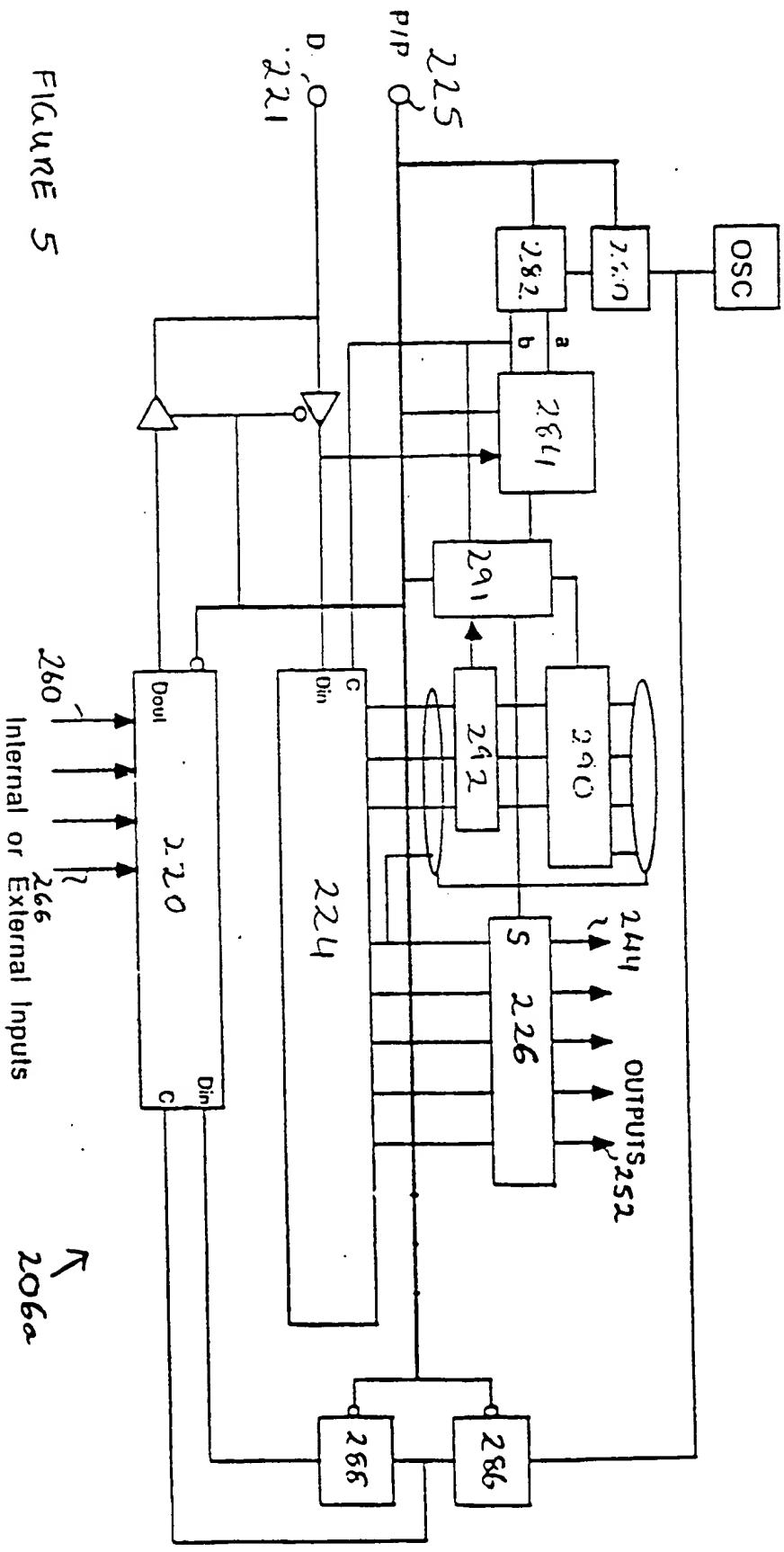


Figure 4

BNSDOCID: <EP 0431434A2>

FIGURE 5



BNSDOCID: <EP 0431434A2>

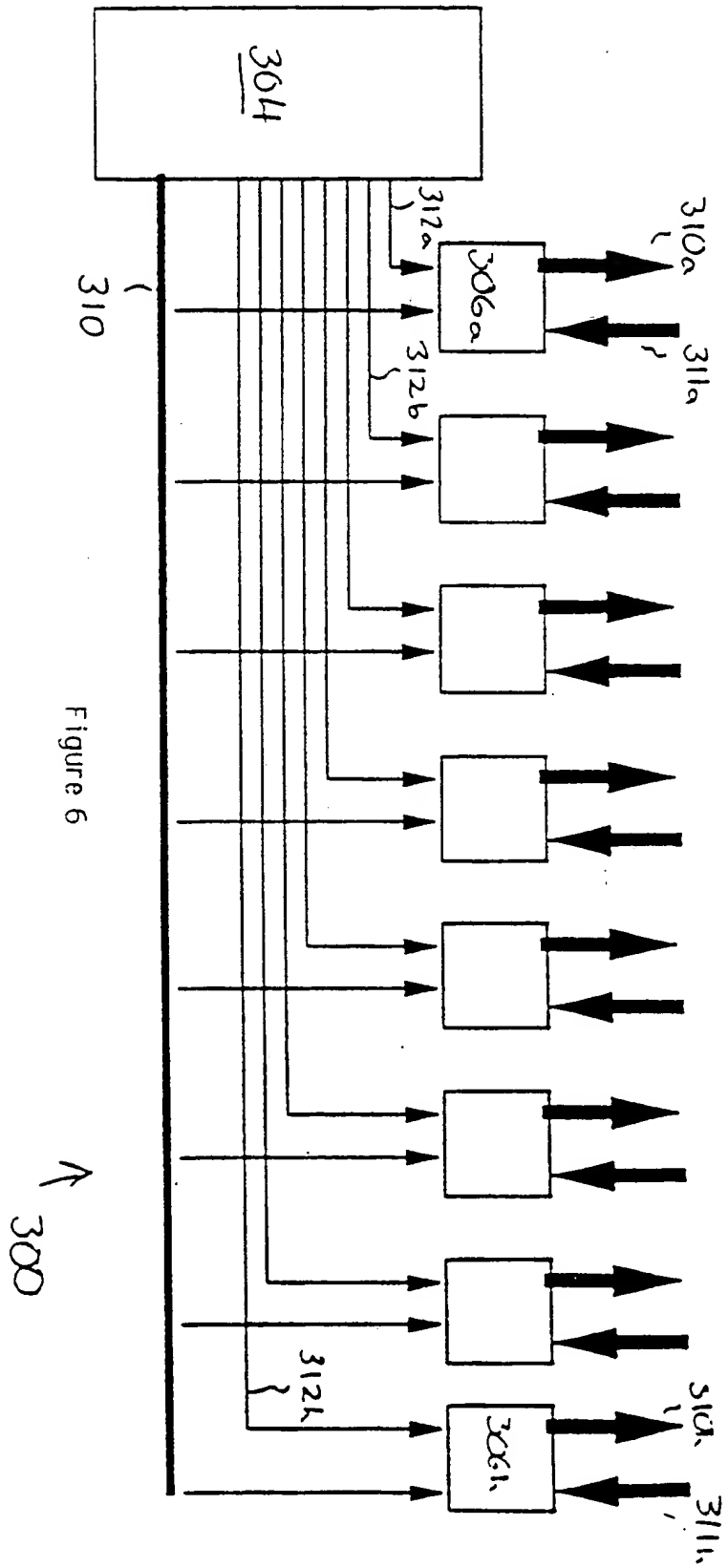
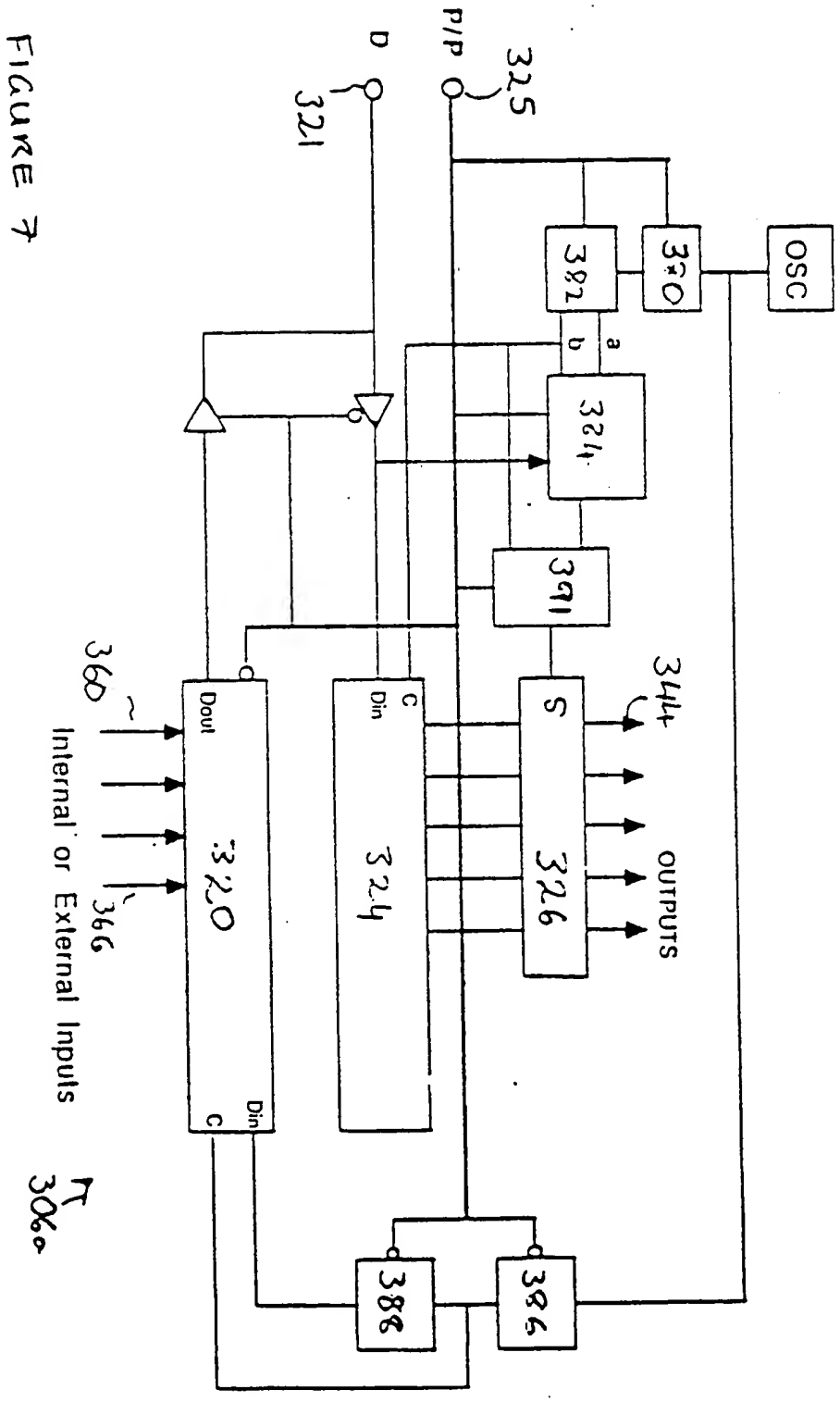


Figure 6

BNSDOCID: <EP 0431434A2>

FIGURE 7



BNSDOCID: <EP 0431434A2>

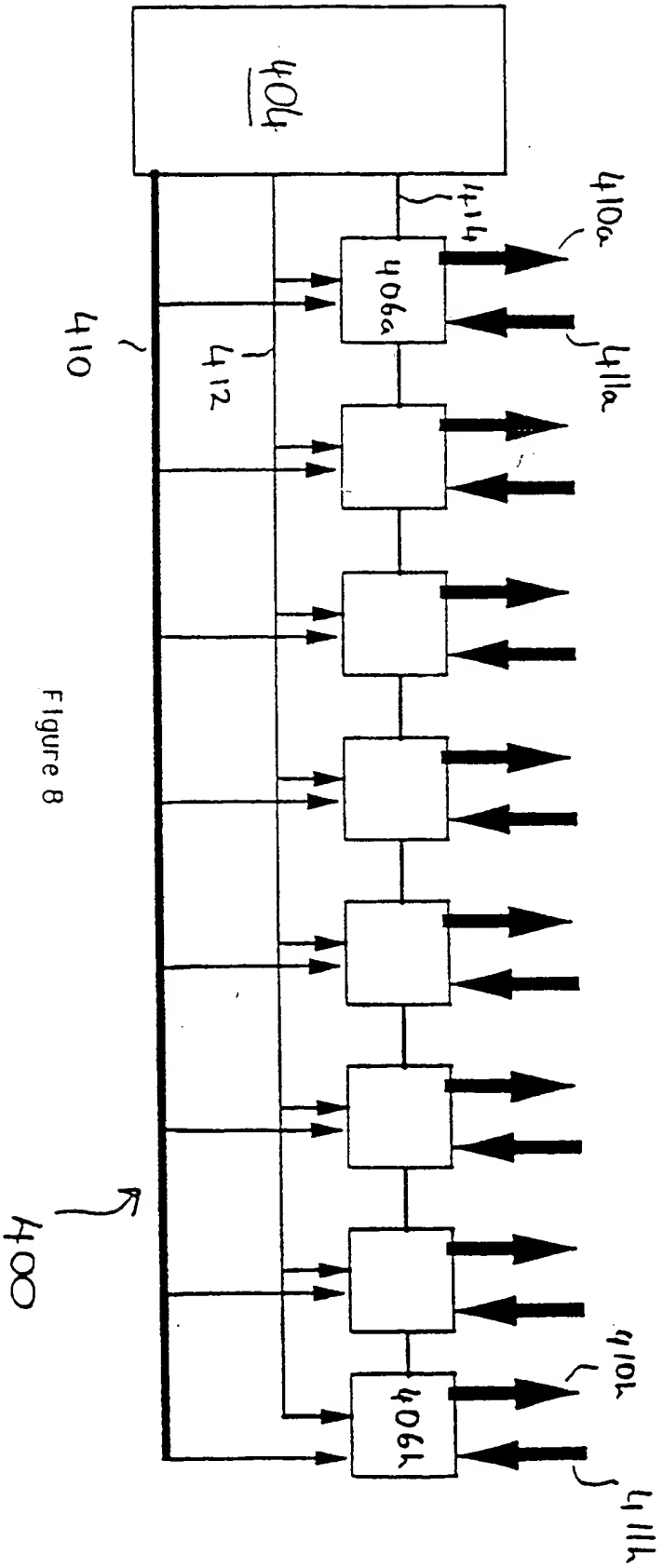


Figure 8

BNSDOCID: <EP 0431434A2>

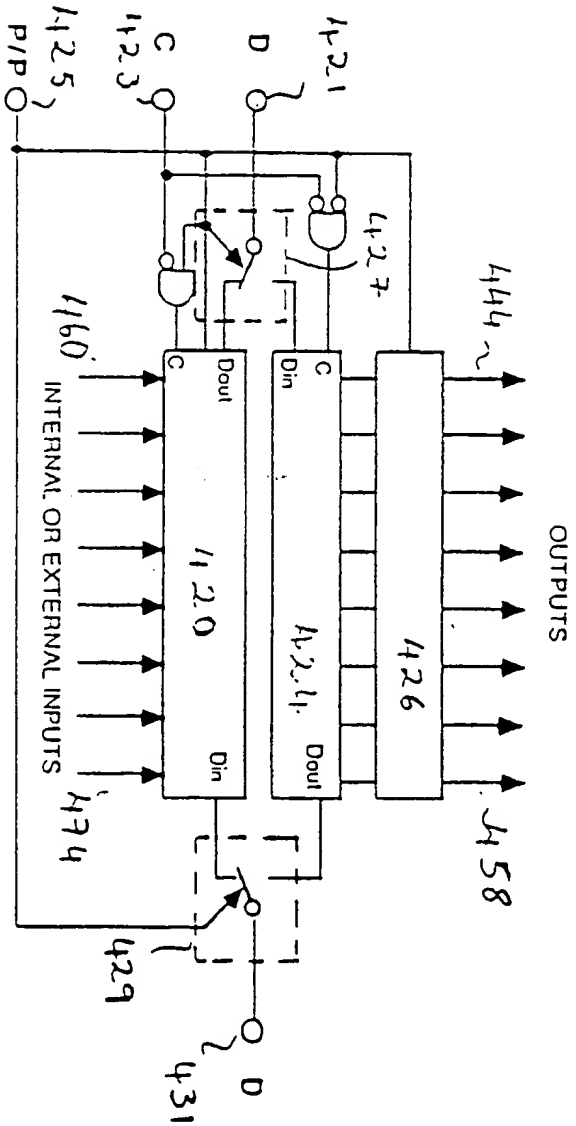


FIGURE 9

1406a

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)